

REMARKS

Please reconsider the present application in view of the above amendments and the following remarks. Applicant thanks the Examiner for acknowledging the substitute declaration filed on December 13, 2004, for accepting the replacement drawings submitted on December 14, 2004, and for carefully considering this application.

Disposition of Claims

Claims 1-27 are pending in the present application. Claims 1, 10, and 19 are independent. The remaining claims depend, directly or indirectly, from claims 1, 10, and 19.

Claim Amendments

Independent claims 1, 10, and 19 have been amended by way of this reply. No new matter has been added by way of these amendments, as support for these amendments may be found, for example, on page 10, lines 4-6 of the present application.

Rejection(s) under 35 U.S.C § 103

Claims 1-27 are rejected under 35 U.S.C. § 103(a) as being obvious over U.S. Patent No. 5,926,814 issued to Fridman et al. (hereinafter "Fridman") in view of "CU Arcadia Project Software System" by Heimbigner (hereinafter "the Heimbigner paper"). Claims 1, 10, and 19 have been amended in this reply to clarify the present invention recited. To the extent that this rejection may still apply to the amended claims, the rejection is respectfully traversed.

The present invention is directed to a lexical analysis parser that is dynamically configurable. Specifically, the lexical analyzer may be dynamically configured at runtime to recognize additional terms (*see, e.g.*, Specification, page 6, lines 1-4; page 10, lines 4-6).

A lexical analyzer in accordance with an embodiment of the present invention divides source text into a set of tokens that are recognized as part of the lexicon of a given language (*see, e.g.,* Specification, page 9, lines 12-14). A lexical analyzer in accordance with an embodiment of the present invention maintains an internal dictionary of recognized tokens (*see, e.g.,* Specification, page 9, lines 16-23). As discussed with reference to Figure 2B, a lexical analyzer is instantiated, and a determination is made as to whether a reserved word or a token is to be added to the internal dictionary of the lexical analyzer (*see, e.g.,* Specification, page 10, lines 8-15). Source code may then be analyzed to determine whether tokens exist in the source code, and to generate the tokens as necessary (*see, e.g.,* Specification, page 10, lines 17-22).

Accordingly, amended independent claim 1 requires obtaining one or more entries and using a computer system to generate a plurality of tokens from a source program at runtime, where the entries may be used to generate a subset of the plurality of tokens, and that at least one of the entries is added at runtime.

Fridman, in contrast to the present invention, fails to teach or suggest at least the above limitations of the claimed invention. Fridman is directed to a method and system for selectively modifying date fields of an application program to solve the year 2000 problem (*see* Fridman, abstract). Date fields in a program are assessed to determine whether the year fields therein have enough space to store 3-digit fields. If the fields do not have enough space, they are modified to accept a 3-digit year that is part of a proprietary date system known as the “Consist Calendar” (*see* Fridman, col. 2, lines 51-60; col. 3, 4-27). As acknowledged by the Examiner, Fridman is completely silent with respect to generating a plurality of tokens at runtime.

As described by Fridman, a token is generated based on a comparison with a static Lexic module and the source code to be modified (*see* Fridman, col. 4, lines 38-56). If the Lexic module of Fridman is used to generate tokens, *all* of the tokens are generated from the (static)

Lexic module. Fridman does not teach or suggest generating tokens from a source program, where entries may be used to generate a subset of the tokens. Fridman is additionally silent with respect to adding entries at runtime.

The Examiner asserts that the Heimbigner paper discloses a runtime lexical analyzer that is able to generate a subset of tokens by virtue of sub-typing. Applicant respectfully disagrees. The Heimbigner paper, as discussed above with reference to Fridman, fails to teach or suggest at least the above limitations of the claimed invention. The Examiner appears to assume that a runtime parsing engine allows tokens to be generated at runtime. However, this merely implies that the runtime parse engine (“yyparse.java”) is executed at runtime.

The Heimbigner paper is directed to a means to take a parser generated using the Gnu Bison parser generator system and translate the parser to execute in Java™. As discussed in the Heimbigner paper, the Java Bison Parser Runtime system (“the jb system”) takes a C output file from Bison and scans it to obtain parse tables and constants. Then, various templates, specified by the user, are filled in using the information extracted from the C output file (*see* the Heimbigner paper, page 1, section labeled “yyparse.java”). The Heimbigner paper is completely silent with respect to generating a plurality of tokens from said source program at runtime using a computer system, where obtained entries are used to generate a subset of the plurality of tokens.

Further the Examiner contends that the Heimbigner paper discloses generating a subset of tokens. The Heimbigner paper states that token classes other than the predefined subclasses may be defined (*see* the Heimbigner paper, page 2, section labeled “token.java”). However, the Heimbigner paper is completely silent with respect to generating tokens from a source program *at runtime*, where obtained entries may be used to generate a subset of the tokens.

Further, the Heimbigner paper states that a subclass of `yylex.java`, the lexical scanner used by the parser, generally imports the `tokentypes` class to get access to the `tokentype` values (see the Heimbigner paper, page 2, section labeled “`yylex.java`”). However, as discussed above, the Heimbigner paper is silent with respect to obtaining any entries defining the tokens at runtime. The Heimbigner paper is completely silent with respect to generating a plurality tokens at runtime, where the entries may be used to generate a subset of the plurality of tokens.

Further, Applicant notes that there is no motivation to combine the cited references. The Examiner cannot combine prior art references to render a claimed invention obvious by merely showing that all the limitations of the claimed invention can be found in the prior art references. There must be a suggestion or motivation to combine the references within the prior art references themselves. In other words, regardless of whether prior art references can be combined, there must be an indication within the prior art references expressing desirability to combine the references. *In re Mills*, 916 F.2d 680 (Fed. Cir. 1990) (emphasis added). Further, the present application *cannot be used as a guide* in reconstructing elements of prior art references to render the claimed invention obvious. *In re Vaeck*, 947 F.2d 488 (Fed. Cir. 1991) (emphasis added).

One skilled in the art would not be motivated by Fridman, which is directed to modifying date fields of a program, to incorporate the teachings of the Heimbigner paper, which is directed to translating a parser generated by the Gnu Bison parser generator to execute in Java™, without the present application as a guide. The Examiner appears to assume that a runtime parsing engine allows tokens to be generated at runtime (based on entries added at runtime). However, this merely implies that the runtime parse engine (“`yyparse.java`”) is executed at runtime. Both Fridman *and* the Heimbigner paper are completely silent with respect to generating tokens at runtime, or other modifications at runtime. Thus, without the present

application as a guide, one skilled in the art would not be motivated to combine Fridman with the Heimbigner paper. Accordingly, the Examiner improperly combined Fridman with the Heimbigner paper in an attempt to render the claimed invention obvious.

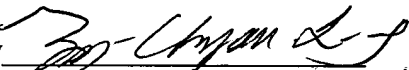

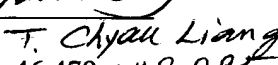
In view of the above, Fridman and Heimbigner, whether considered separately or in combination, (i) fail to show or suggest the claimed invention as recited in amended independent claims 1, 10, and 19 and (ii) are not properly combinable. Thus, amended independent claim 1 is patentable over Fridman and Heimbigner. Dependent claims 2-9 are allowable for at least the same reasons. Independent claims 10 and 19 are rejected for the same reasons as claim 1 and include essentially the same subject matter as claim 1. Thus, independent claims 10 and 19 are also patentable over Fridman and Heimbigner. Dependent claims 11-18 and 20-27 are allowable for at least the same reasons. Accordingly, withdrawal of this rejection is respectfully requested.

Conclusion

Applicant believes this reply is fully responsive to all outstanding issues and places the present application in condition for allowance. If this belief is incorrect, or other issues arise, the Examiner is encouraged to contact the undersigned or his associates at the telephone number listed below. Please apply any charges not covered, or any credits, to Deposit Account 50-0591 (Reference Number 16159/096001; P5942).

Dated: August 15, 2005

Respectfully submitted,

By 
for  Robert P. Lord 
Registration No.: 46,479 #48,885
OSHA · LIANG LLP
1221 McKinney St., Suite 2800
Houston, Texas 77010
(713) 228-8600
(713) 228-8778 (Fax)
Attorney for Applicant